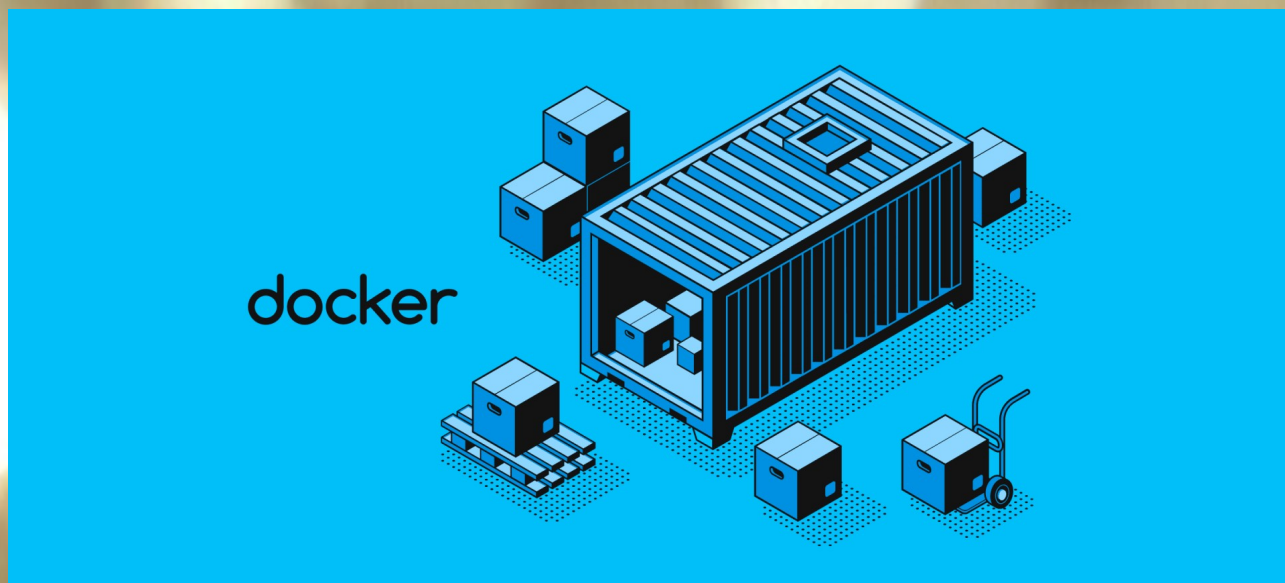


Luis Fernando Valverde Cárdenas
jueves, 4 de enero de 2024

IES Cura Valera



PROYECTO HLC: DOCKERSRI



Profesor: José Antonio Morgado Berruezo

Curso: 2023/2024 (2º ASIR)

Técnico Superior en Administración de Sistemas Informáticos en Red

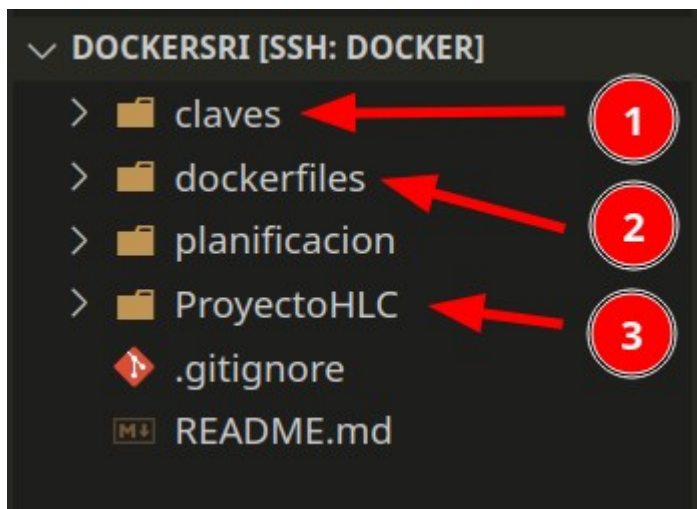
Módulo: Horas de libre configuración (HLC)

ÍNDICE

Introducción.....	1
Dockerfiles.....	1
debian_base.....	2
Dockerfile.....	2
Scripts.....	2
dhcp_hlc.....	3
Dockerfile.....	3
Scripts.....	4
Archivo de configuración.....	4
dns_hlc.....	5
Dockerfile.....	6
Scripts.....	6
Archivos de configuración.....	7
web_log.....	8
Scripts.....	8
Archivos de configuración.....	9
ProyectoHLC.....	9
Despliegue.....	10
Datos de los servicios.....	10
dhcp_hlc.....	10
dns_hlc.....	11
logs.....	12

Introducción

La estructura de este proyecto está formado por los siguientes directorios:



1) claves: Directorio donde se almacenan las claves rsa que se copiarán al contenedor para acceder si contraseña.

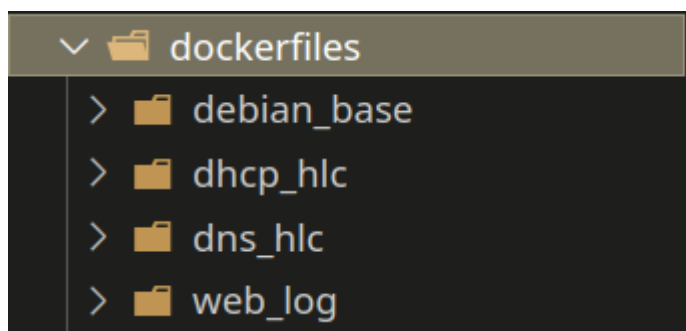
2) dockerfiles: Directorio donde se construyen las imágenes de los servicios

3) ProyectoHLC: Directorio donde están los volúmenes con los datos de los servicios, docker-compose.yml y el archivo de entorno para el despliegue del proyecto.

Extra) planificacion: Directorio que contiene un esquema que muestra el entorno y puesta en marcha del proyecto.

Dockerfiles

En el apartado de dockerfiles como se ha dicho anteriormente se produce la construcción de las imágenes de cada servicio:



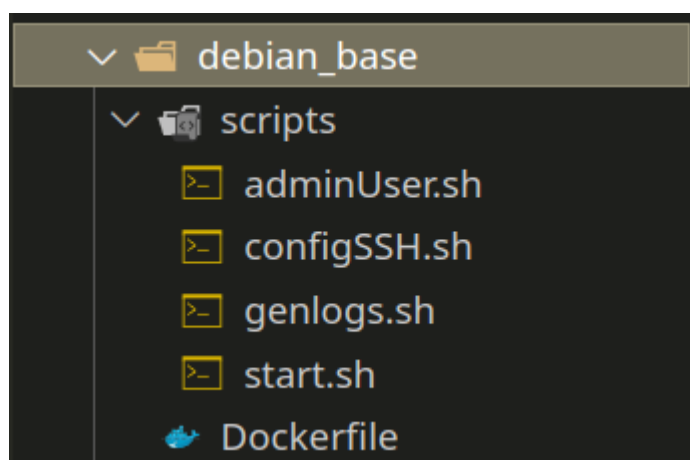
debian_base

Esta es la imagen base (basada en debian) donde partirán los servicios del proyecto.

Esta imagen dispondrá del servicio SSH para la administración remota, un usuario administrador distinto de root para la administración y conexión remota de forma más segura.

Para la construcción de la imagen se utilizará el archivo Dockerfile (construcción del sistema de archivos del contenedor) y la carpeta scripts que contendrá los archivos de configuración (servicios, usuarios...)

Dockerfile



Dockerfile: Construiremos el sistema de archivos del contenedor a partir de una imagen debian donde:

1. Establecer zona horaria.
2. Actualización y instalación de paquetes base como openssh-server, vim, iptables, iputils-ping...
3. Creación de los directorios scripts y logs en **/root**

4. Copiado de el contenido de la carpeta scripts de la imagen al directorio del contenedor **/root/scripts**

5. Exponer puerto SSH

Scripts

AdminUser.sh → Creación de un usuario administrador pero antes hay que comprobar de que no exista el usuario comprobando el archivo **/etc/passwd** y de que no exista su home

configSSH.sh → Comprobamos de que exista el fichero de configuración SSH **/etc/ssh/sshd_config**, si existe procedemos modificarlo para que el puerto sea 2222 en vez de 22 por ejemplo y de que esté prohibido acceder vía SSH al usuario root

Comprobamos de que exista el directorio **/var/run/sshd** y si no es así lo creamos.

Comprobamos de que exista el home del usuario administrador creado anteriormente, si es así comprobamos de que exista dentro el directorio **.ssh**, si no es así lo creamos y aquí redireccionamos el contenido de la clave rsa copiada en el contenedor al archivo **authorized_keys**

Y finalmente levantamos el servicio SSH en segundo plano.

genlogs.sh → Este script contiene funciones para generar un archivo html que será el formato en los que se visualizará los logs que se hayan producido en cada una de comprobaciones o acciones de configuración a través de mensajes de éxito o error y su hora en formato tabla.

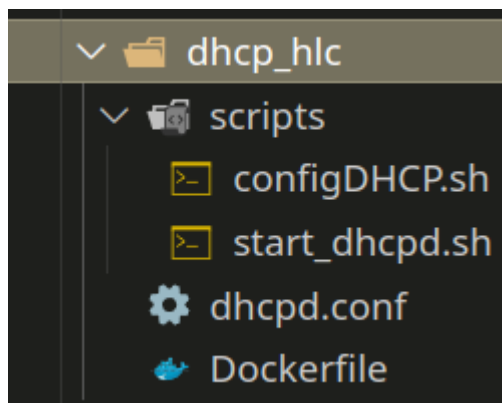
start.sh → Script principal de configuración que carga y utiliza las funciones de los scripts anteriores y comprueba de manera general y resumida los resultados de cada una de las funciones y finaliza la configuración.

dhcp_hlc

Imagen construida a partir de la imagen base que contiene el servicio DHCP.

Para la construcción de la imagen se utilizará el archivo Dockerfile (construcción del sistema de archivos del contenedor), la carpeta scripts que contendrá los archivos de configuración (servicios, usuarios...) y el archivo de configuración del servicio DHCP para este proyecto.

Dockerfile



Dockerfile: Construiremos el sistema de archivos del contenedor a partir de una imagen debian donde:

1. Actualización y instalación de los paquetes necesarios para el servicio DHCP
2. Limpiado de cache y dependencias huérfanas.
3. Copiado de la clave rsa al directorio **/root**
4. Copiado de el contenido de la carpeta scripts de la imagen al directorio del contenedor **/root/scripts** y además le daremos permisos de ejecución.
5. Copiado del archivo de configuración DHCP en el contenedor en **/etc/dhcp**
6. Exponer puerto DHCP
7. Ejecución del script de inicialización de la configuración del servicio DHCP

Scripts

ConfigDHCP.sh → Comprobamos de que exista el fichero `/etc/default/isc-dhcp-server`, si es así añadimos con sed las interfaces de red que actuarán en el servicio DHCP la cuál solo es una la que tiene el contenedor

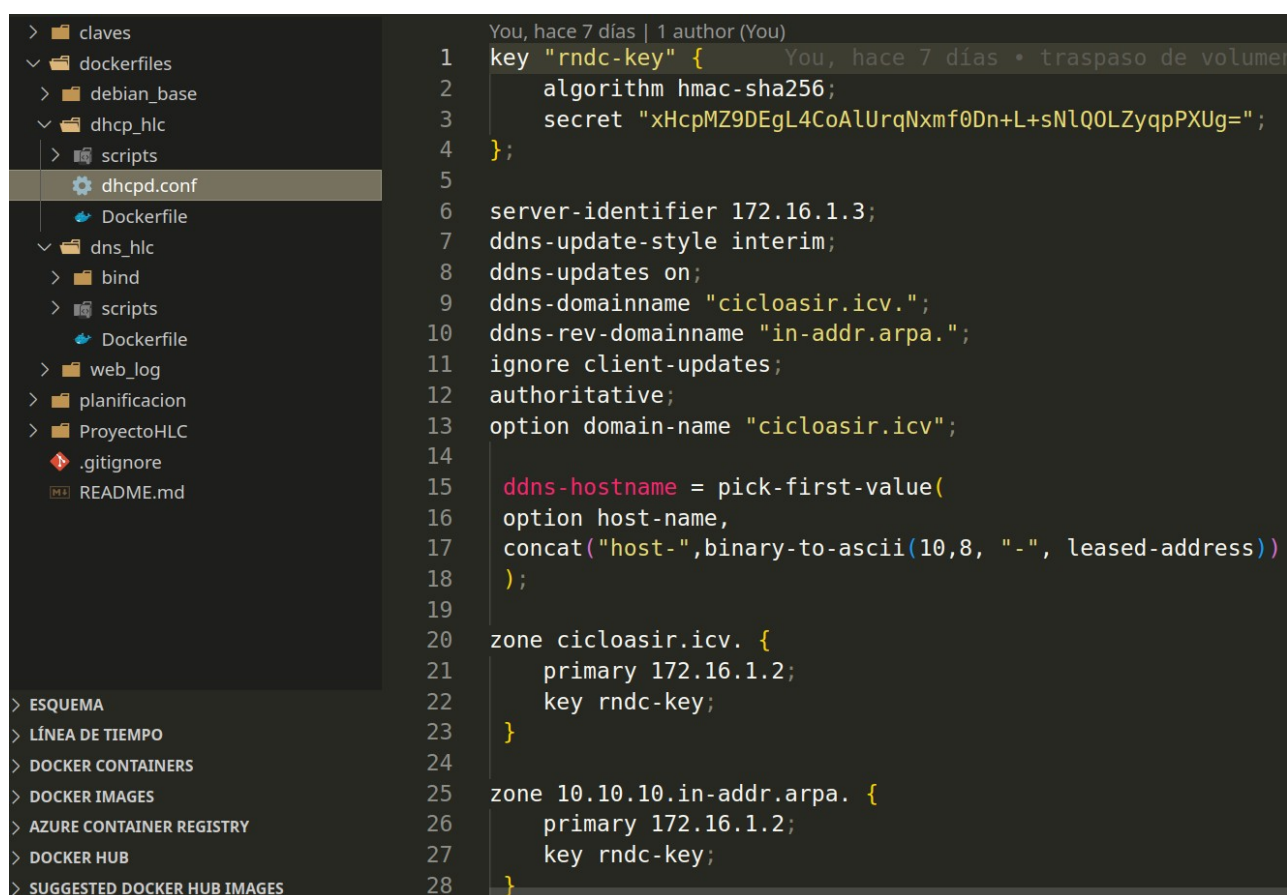
Comprobamos de que exista el usuario `dhcpd` en `/etc/passwd`, si no es así procedemos a crearlo

Comprobamos de que exista los ficheros de concesiones en `/var/lib/dhcp`, si no existen se deben crear vacíos.

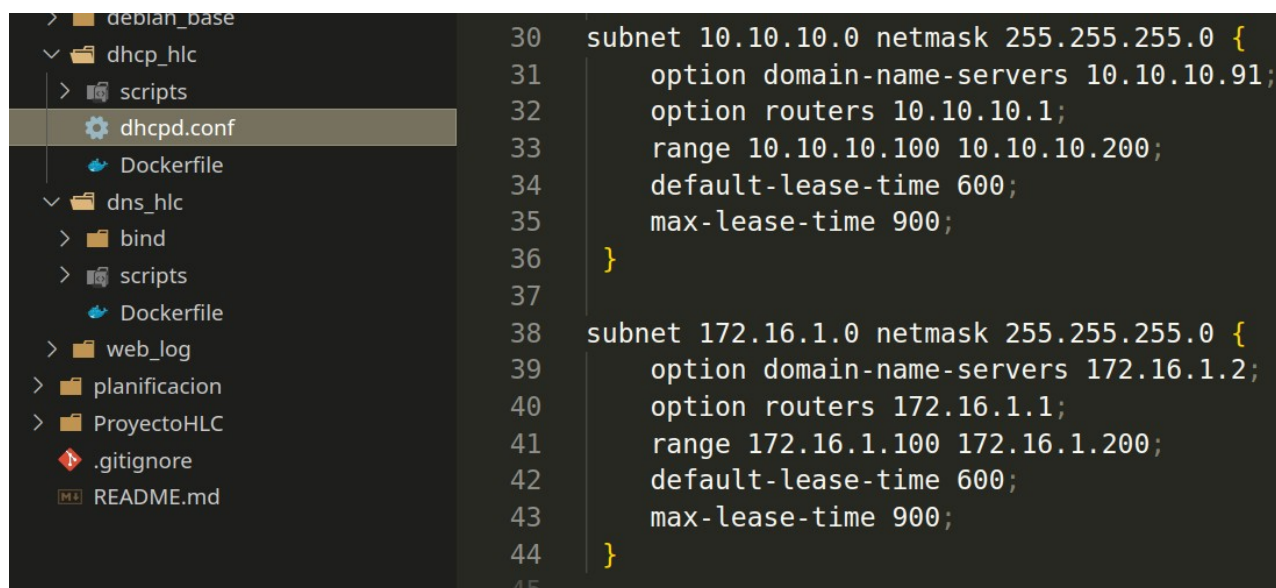
Asegurar de que el directorio `/var/lib/dhcp` es propiedad del usuario `dhcpd` (servicio) para que pueda escribir.

start_dhcpd.sh → Script de configuración que carga y utiliza las funciones de los archivos anteriores y comprueba de forma general y resumida de que está todo correcto para levantar el servicio DHCP

Archivo de configuración.



```
1 key "rndc-key" {
2     algorithm hmac-sha256;
3     secret "xHcpMZ9DEgL4CoA1UrqNxmf0Dn+L+sNlQOLZyqpPXUg=";
4 };
5
6 server-identifier 172.16.1.3;
7 ddns-update-style interim;
8 ddns-updates on;
9 ddns-domainname "cicloasir.icv.";
10 ddns-rev-domainname "in-addr.arpa.";
11 ignore client-updates;
12 authoritative;
13 option domain-name "cicloasir.icv";
14
15 ddns-hostname = pick-first-value(
16     option host-name,
17     concat("host-",binary-to-ascii(10,8, "-", leased-address))
18 );
19
20 zone cicloasir.icv. {
21     primary 172.16.1.2;
22     key rndc-key;
23 }
24
25 zone 10.10.10.in-addr.arpa. {
26     primary 172.16.1.2;
27     key rndc-key;
28 }
```



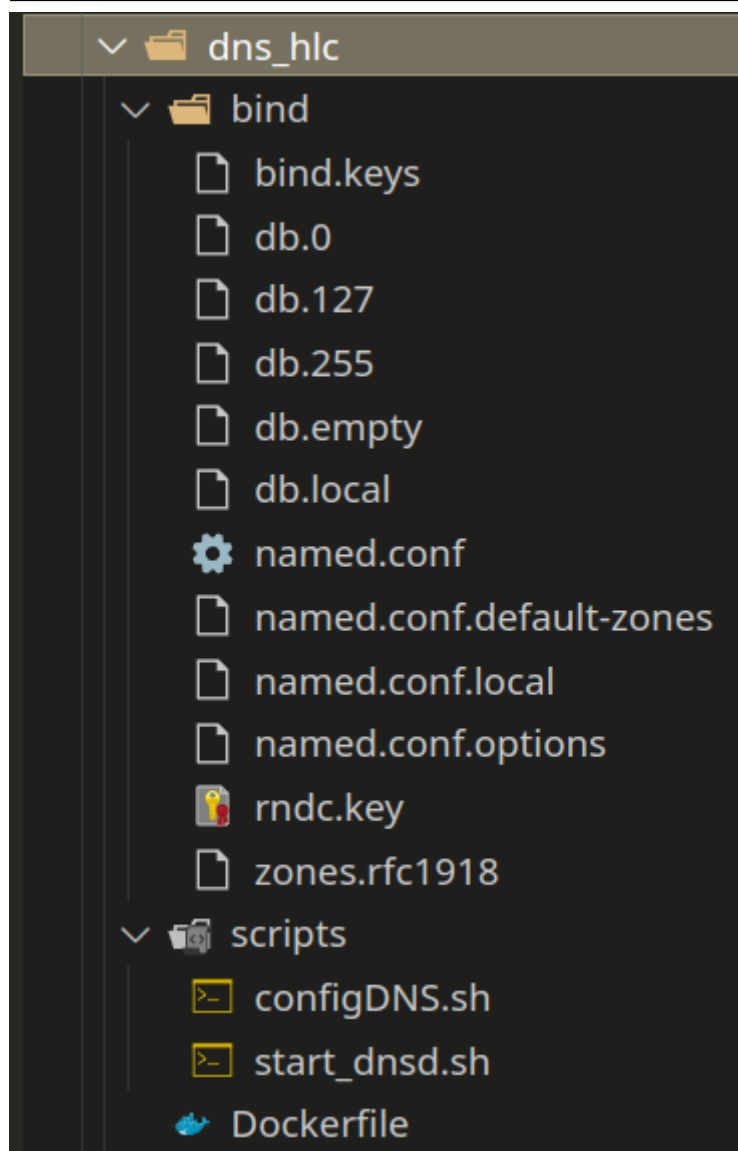
```
30 subnet 10.10.10.0 netmask 255.255.255.0 {
31     option domain-name-servers 10.10.10.91;
32     option routers 10.10.10.1;
33     range 10.10.10.100 10.10.10.200;
34     default-lease-time 600;
35     max-lease-time 900;
36 }
37
38 subnet 172.16.1.0 netmask 255.255.255.0 {
39     option domain-name-servers 172.16.1.2;
40     option routers 172.16.1.1;
41     range 172.16.1.100 172.16.1.200;
42     default-lease-time 600;
43     max-lease-time 900;
44 }
```

dns_hlc

Imagen construida a partir de la imagen base que contiene el servicio DNS.

Para la construcción de la imagen se utilizará el archivo Dockerfile (construcción del sistema de archivos del contenedor), la carpeta scripts que contendrá los archivos de configuración (servicios, usuarios...) y la carpeta con la configuración del servicio DNS.

Dockerfile



Dockerfile: Construiremos el sistema de archivos del contenedor a partir de una imagen debian donde:

1. Actualización y instalación de los paquetes necesarios para el servicio DNS
2. Limpiado de cache y dependencias huérfanas.
3. Copiado de la clave rsa al directorio **/root**
4. Copiado de el contenido de la carpeta scripts de la imagen al directorio del contenedor **/root/scripts** y además le daremos permisos de ejecución.
5. Copiado del directorio de configuración DNS en el contenedor en **/etc/bind**
6. Exponer puertos DNS
7. Ejecución del script de inicialización de la configuración del servicio DNS

Scripts

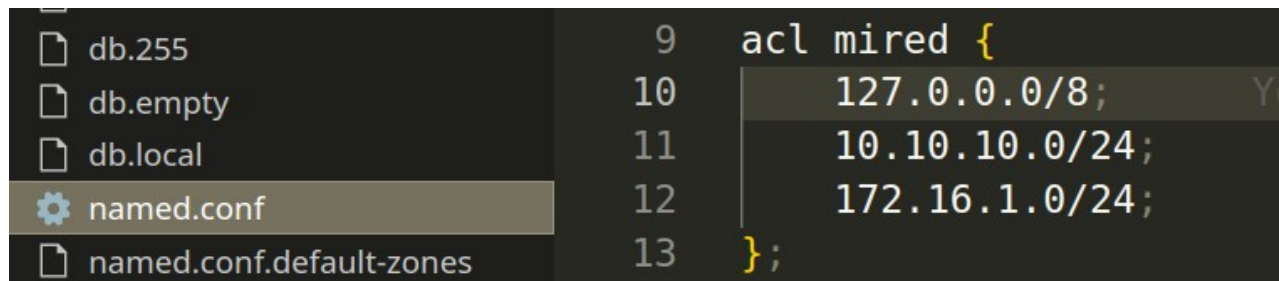
ConfigDNS.sh → Asegurar de que los directorios **/etc/bind** y **/var/cache/bind** son propiedad del usuario bind, si no es así cambiaremos el propietario.

Realizar un chequeo de zonas para comprobar de que las zonas estén configuradas correctamente.

start.dnssd.sh → Script de configuración que carga y utiliza las funciones de los archivos anteriores y comprueba de forma general y resumida de que está todo correcto para levantar el servicio DNS.

Archivos de configuración.

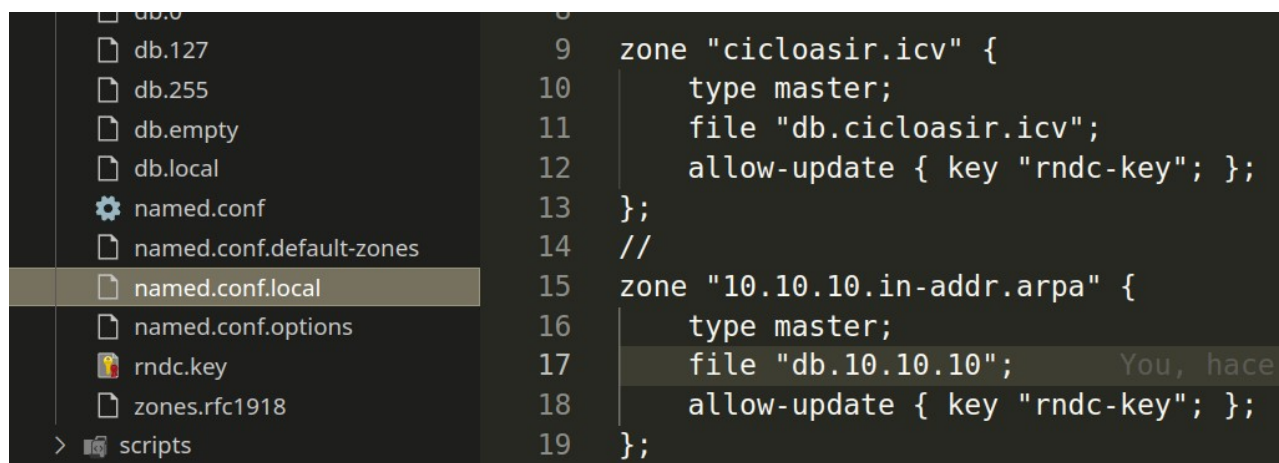
Lista acl definida para restringir acceso a recursos que no están especificados en el rango



```

9  acl mired {
10      127.0.0.0/8;
11      10.10.10.0/24;
12      172.16.1.0/24;
13  };
  
```

Zonas directas e inversas definidas junto a la ubicación de sus respectivos archivos de zonas, el tipo de zona y la clave para la comunicación segura entre el servidor DNS y el servidor DHCP



```

9  zone "cicloasir.icv" {
10      type master;
11      file "db.cicloasir.icv";
12      allow-update { key "rndc-key"; };
13  };
14  //
15  zone "10.10.10.in-addr.arpa" {
16      type master;
17      file "db.10.10.10";
18      allow-update { key "rndc-key"; };
19  };
  
```

Especificar la clave,acl y la sentencia controls



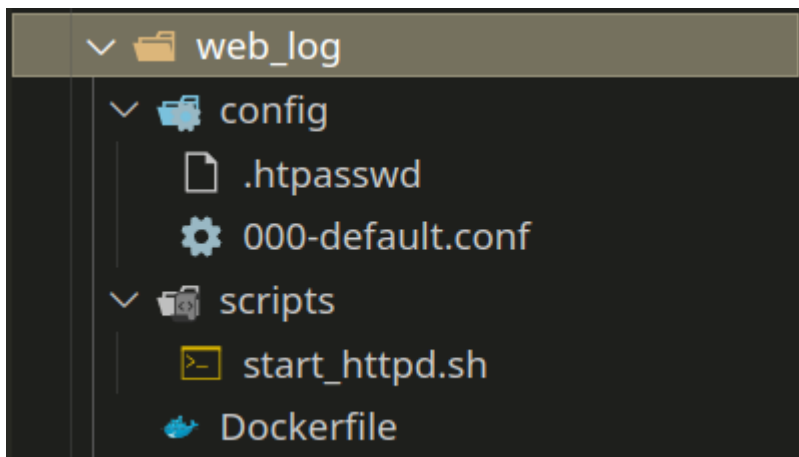
```

24  listen-on-v6 { any; };
25  allow-query { mired; };
26  };
27
28  key "rndc-key" {
29      algorithm hmac-sha256;
30      secret "xHcpMZ9DEgL4CoAlUrqNxmf0Dn+L+sNlQ0LZyqpPXUg=";
31  };
32
33  controls {
34      inet 127.0.0.1 allow { 127.0.0.1; 172.16.1.3; 10.10.10.91; 172.16.1.2; } keys
35      { rndc-key; };
36  };
  
```

web_log

Imagen construida a partir de la imagen base que contiene el servicio HTTP.

Para la construcción de la imagen se utilizará el archivo Dockerfile (construcción del sistema de archivos del contenedor), la carpeta scripts que contendrá los archivos de configuración (servicios, usuarios...), el archivo de configuración del servicio HTTP un archivo con los usuarios y contraseñas para el autenticado en la web para este proyecto.



Dockerfile: Construiremos el sistema de archivos del contenedor a partir de una imagen debian donde:

1. Actualización y instalación de los paquetes necesarios para el servicio HTTP
2. Limpiado de cache y dependencias huérfanas.

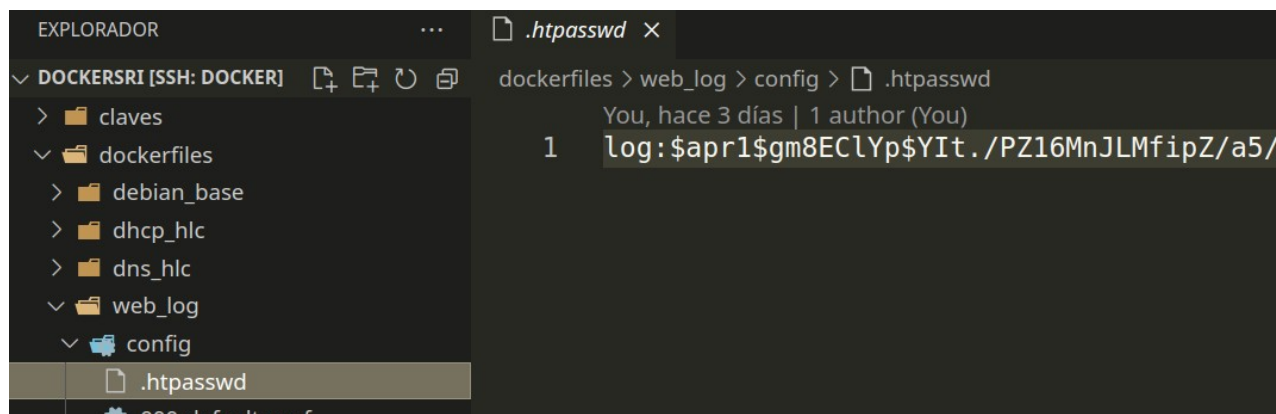
3. Copiado de la clave rsa al directorio **/root**
4. Copiado de el contenido de la carpeta scripts de la imagen al directorio del contenedor **/root/scripts** y además le daremos permisos de ejecución.
5. Copiado del archivo de configuración al directorio **/etc/apache2/sites-available** y del archivo usuarios y contraseñas a **/etc/apache2**
6. Exponer puertos HTTP
7. Ejecución del script de inicialización de la configuración del servicio HTTP

Scripts

start_httpd.sh → Script de configuración que carga y utiliza las funciones de los archivos anteriores y comprueba de forma general y resumida de que está todo correcto para levantar el servicio HTTP.

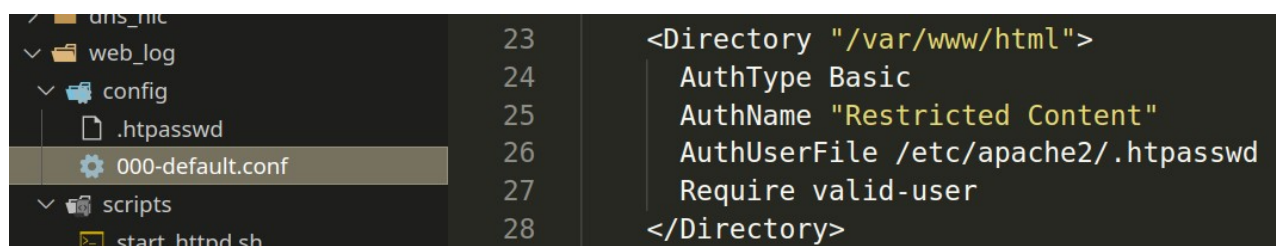
Archivos de configuración.

Archivo con el usuario y contraseña para la autenticación en el servicio web.



The screenshot shows a file explorer window titled "EXPLORADOR" with a sidebar listing folders like "claves", "dockerfiles", "debian_base", "dhcp_hlc", "dns_hlc", "web_log", and "config". The "config" folder is expanded, showing the ".htpasswd" file. The main pane displays the content of ".htpasswd" with a single line: "1 log:\$apr1\$gm8EClyp\$YIt./PZ16MnJLMfipZ/a5/". Above the line, a status bar indicates "You, hace 3 días | 1 author (You)".

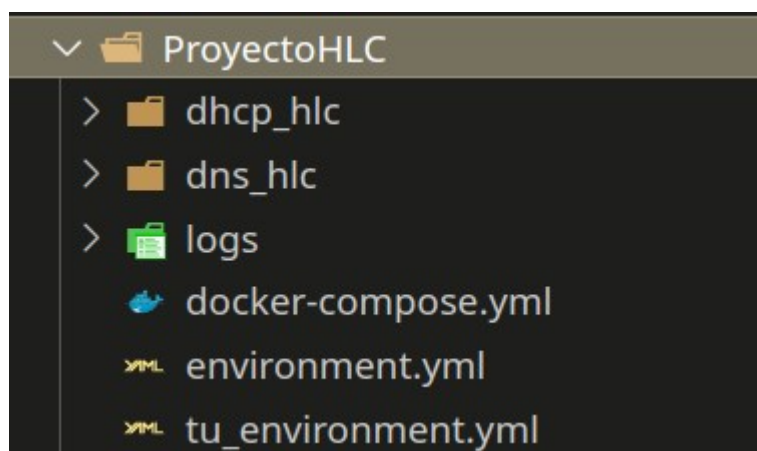
Definimos las restricciones de la página a través del archivo .htpasswd



The screenshot shows a file explorer window with a sidebar listing folders like "dns_hlc", "web_log", "config", and "scripts". The "config" folder is expanded, showing the ".htpasswd" file. The main pane displays the content of ".htpasswd" with a single line: "1 log:\$apr1\$gm8EClyp\$YIt./PZ16MnJLMfipZ/a5/". Above the line, a status bar indicates "You, hace 3 días | 1 author (You)".

ProyectoHLC.

En el directorio ProyectoHLC es donde sucede el despliegue de los servicios y donde se almacena los volúmenes con los datos de los servicios.

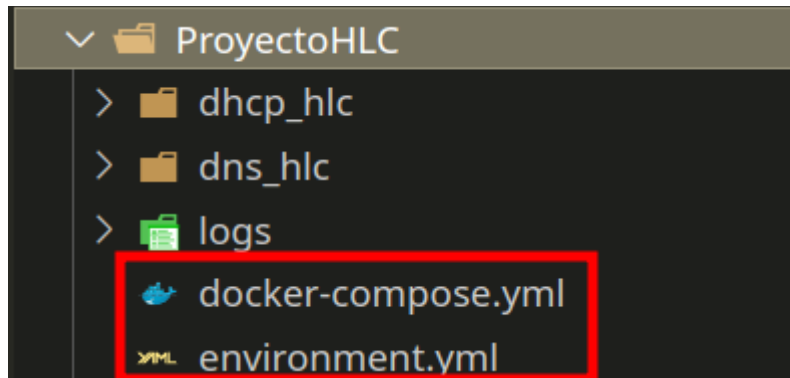


Despliegue.

Estos dos archivos orquestan el despliegue de los servicios del proyecto

docker-compose.yml → Archivo que utiliza las variables del environment para desplegar los servicios especificando puertos, volúmenes, imágenes...

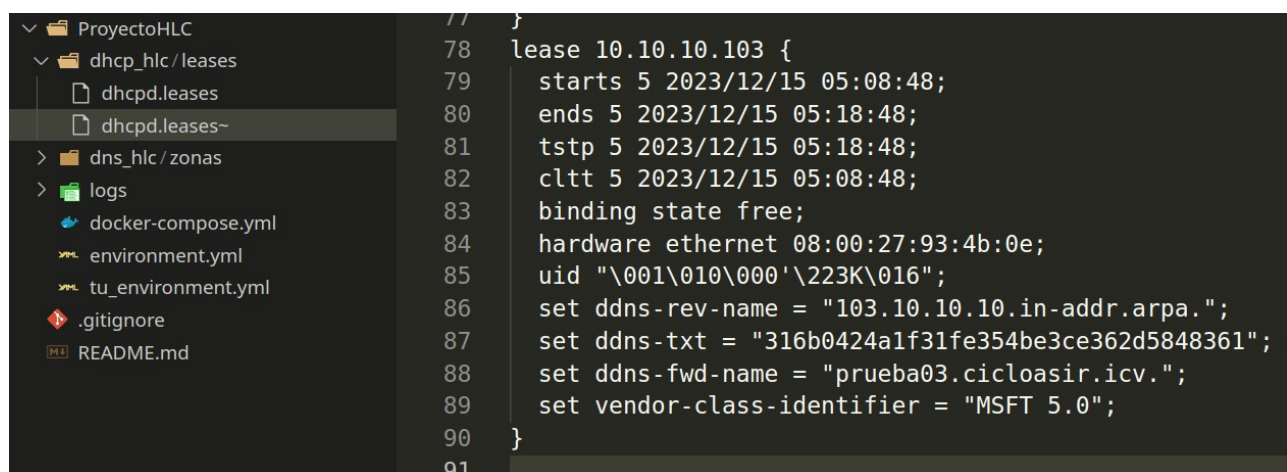
environment.yml → Archivo que almacena las variables de entorno para el despliegue.



Datos de los servicios.

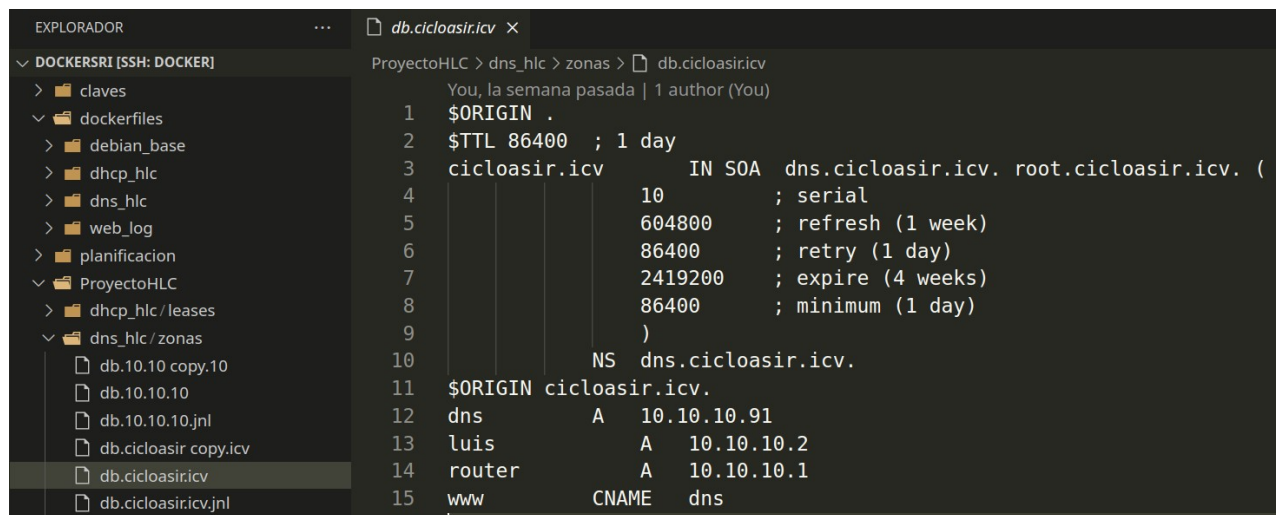
dhcp_hlc

En la carpeta **leases** se almacenan las concesiones de las ips a los host por parte de servidor DHCP.

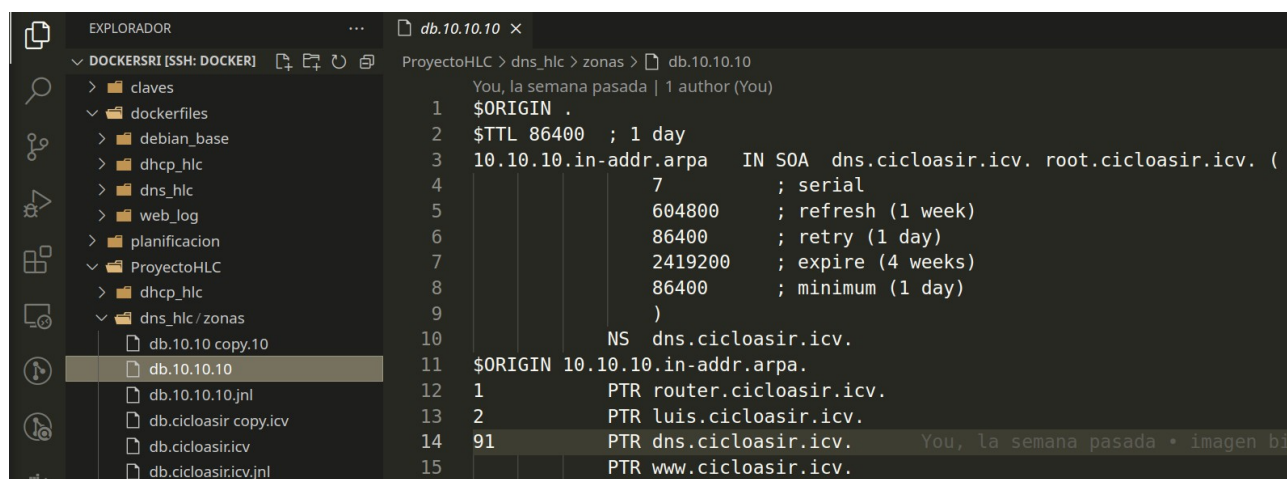


dns_hlc

En la carpeta **zonas** almacenamos los archivos que contienen definidas los registros de las zonas inversas y directas.



```
EXPLORADOR
...
db.cicloasir.icv x
ProyectoHLC > dns_hlc > zonas > db.cicloasir.icv
You, la semana pasada | 1 author (You)
1 $ORIGIN .
2 $TTL 86400 ; 1 day
3 cicloasir.icv IN SOA dns.cicloasir.icv. root.cicloasir.icv. (
4 10 ; serial
5 604800 ; refresh (1 week)
6 86400 ; retry (1 day)
7 2419200 ; expire (4 weeks)
8 86400 ; minimum (1 day)
9 )
10 NS dns.cicloasir.icv.
11 $ORIGIN cicloasir.icv.
12 dns A 10.10.10.91
13 luis A 10.10.10.2
14 router A 10.10.10.1
15 www CNAME dns
```



```
EXPLORADOR
...
db.10.10.10 x
ProyectoHLC > dns_hlc > zonas > db.10.10.10
You, la semana pasada | 1 author (You)
1 $ORIGIN .
2 $TTL 86400 ; 1 day
3 10.10.10.in-addr.arpa IN SOA dns.cicloasir.icv. root.cicloasir.icv. (
4 7 ; serial
5 604800 ; refresh (1 week)
6 86400 ; retry (1 day)
7 2419200 ; expire (4 weeks)
8 86400 ; minimum (1 day)
9 )
10 NS dns.cicloasir.icv.
11 $ORIGIN 10.10.10.in-addr.arpa.
12 1 PTR router.cicloasir.icv.
13 2 PTR luis.cicloasir.icv.
14 91 PTR dns.cicloasir.icv.
15 PTR www.cicloasir.icv.
```

logs

En la carpeta **logs** se almacena los logs en archivos html con cada uno de los nombres de cada contenedor

